128 KDE © 1981

SOFT DISK

by

Legend Industries Ltd.

Legend 128KDE Operation Manual

Legend Industries, Ltd.
P.O. Box 112
Pontiac, Mi. 48056

NOTICE

Legend Industries, Ltd. reserves the right to make improvements in this manual and the product described within this manual at any time and without notice.

UPDATE POLICY

Improved or Updated versions of the Software described in this manual will be made available to the original purchaser for the difference in retail price of the new Software, if any, or a minimum $8.00 charge to cover the shipping and handling expenses. Please send your original diskette and payment when requesting a software update to Legend Industries at the address at the bottom of this page.

DISCLAIMER

Legend Industries, Ltd. makes no warranties, either expressed or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability or fitness for any particular purpose. Legend Industries, Ltd. software is sold or licensed "as is". The entire risk as to its quality and performance is with the buyer. Should the programs prove defective following their purchase, the buyer and not Legend Industries, Ltd. (or its distributor or retailer) assumes the entire cost of all necessary servicing, repair or correction and any incidental or consequential damages. In no event will Legend Industries, Ltd. be liable for direct, indirect, incidental or consequential damages resulting from any defect in the software, even if Legend Industries, Ltd. has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages so the above limitation or exclusion may not apply to you.

COPYRIGHT NOTICE

Legend Industries, Ltd.
2220 Scott Lake Road
Pontiac, Mi. 48054

Table of Contents

Introduction 1

Installation 3

Theory of Operation 7

Memory Master 17

Memory Master Listing 22

Disk Emulator 25

Turnkey Emulator Listing 33

Disk Emulator Listing 36

Special Format 41

Floppy Formatter Listing 42

Firmware Selector 43

Firmware Select Listing 44

Introduction

The Legend 128KDE is a continuation in state-of-the-art memory technology for the Apple II Computer. The 128KDE Soft Disk uses "state-of-the-art" 64K RAM chips and requires only one slot to emulate a complete, almost instant access, floppy disk drive. With no motor speed delay of the disk drive to contend with, the 128KDE is as much as 300% faster than the standard Apple disk II drive. The 128KDE can be put into any slot and can be accessed with the standard Apple DOS 3.3 disk drive commands, for example; LOAD, SAVE, READ, WRITE, etc.

The Legend 128KDE card conforms to the same Device Select Address conventions used by Apple Computer Inc. and consequenty will act exactly as an Apple Language Card when installed in slot Ø and used with the Apple UCSD Pascal, Apple FORTRAN, Microsoft CP/M or other systems which utilize a 16K RAM card. The 128KDE is functionally equivalent to two (2) 64KC cards on one (1) card. By bank switching eight (8) 16K banks over the existing ROM space, the 128KDE Soft Disk enables your Apple to triple it's RAM capacity.

A limitation of microcomputers has been the lack of an appropriate amount of RAM storage. The barriers to this limitation have been broken with the introduction of the Legend Industries 64KC and 128KDE cards. With this advancement in state-of-the-art technology the Apple can now address up to 944K of RAM memory with the appropiate software.

The Soft Disk software presently available supports up to four (4) 128KDE cards, providing up to 512K bytes of storage equivalent to four (4) fast access disk drives. A new DOS "MOUNT" command enables the user to MOUNT a complete Floppy Diskette on to one (1) of the emulated disk drives. A new DOS "UPDATE" command copies the contents of the specified emulated disk back onto floppy diskette. This usually takes less than eighteen (18) seconds.

The 128KDE Soft Disk has a short 16 conductor cable which is used to access timing signals on the Apple mother board. The cable plugs into any one of the eight (8) 4116 RAM sockets located at row "E" on the mother board. The RAM chip that normally occupies this socket is removed and reinserted onto the 128KDE card.

Legend Industries, Ltd. encourages the development of sophisticated system software by OEM's which supports the Legend 128KDE system. The engineering staff will be happy to evaluate any software that is developed for use with the 128KDE system on a non-disclosure basis. Please feel free to contact the engineering staff at Legend Industries Ltd.

*Apple II, Apple UCSD Pascal, Apple FORTRAN and Apple Language Card are products of Apple Computer Inc.
*UCSD Pascal is a trademark of Regents U.C.S.D.
*Microsoft CP/M is a product of Microsoft Inc.

Introduction (con't.)

The 128KDE adds 128K of RAM (random access memory) to the Apple Computer by bank switching 16K banks over the existing ROM (read only memory) space. This memory is NOT directly addressable, that is to say you must bank switch to take advantage of the 128K of RAM on the board. Bank switching is the process of selecting one particular 16K bank of RAM or ROM to either read or write and is covered in the Theory of Operation section in this manual. Almost any information can be stored in the 128KDE with the appropriate software. Please read this manual before using the card so you are more familiar with the card and the necessary software to use the card to it's fullest extent. Legend Industries is always developing software for it's products and as Legend Industries and other software houses make available programs using the 64/128K RAM cards, the card will prove to be one of the best investments you ever made.

We at Legend Industries encourage you to experiment with the card and hope you enjoy using the card as much as we at Legend do.

Installation

The 128KDE card can be installed in any slot from number Ø next to the power supply through number 7 nearest the game connector socket. The slot number that you choose is really dictated by the type of peripheral cards that already populate the peripheral connectors on the mother board and the languages and operating systems that you use as well as the intended use of the 128KDE card. The Apple Disk Controller, for example, is almost always installed in slot 6 while a printer card is almost universally installed in slot 1.

If you choose to install the 128KDE in slot Ø then the card will be recognized as a 16K RAM card when used with Apple DOS 3.3. Bank #Ø on the 128KDE will be loaded with the language not contained in ROM on the Apple mother board when you initially 'Boot-up' the DOS 3.3 System Master diskette, making both languages available to the user. Banks 1 thru 7 on the 128KDE card still provide 112K of available memory in addition to the 48K RAM memory on the Apple mother board. Pascal and Microsoft CP/M will also recognize the 128KDE as a RAM board when it is installed in slot Ø.

If you program entirely in BASIC or assembly language and you have an Integer or Applesoft ROM card installed in slot Ø, then you may decide to install the 128KDE card in slot 4, or any other slot. This provides you with both languages available in ROM as well as the full 128K bytes in the 128KDE card in addition to the 48K RAM on the Apple mother board providing 176K of usable RAM memory.

If you have an Apple Language Card or similar 16K RAM card then you have a choice. You can install the 128KDE in slot Ø and the RAM card in some other slot (slot 4 is a good choice) or leave the RAM card in slot Ø and install the 128KDE card in another slot. Either way you have a total of 192K RAM memory. You could also leave the RAM card in slot Ø and install two 128KDE cards in slots 4 and 5 (any slots will do) for a total of 320K of RAM memory and use the DISK EMULATOR program on the demo disk to simulate two almost 'instant access' disk drives in BASIC. The 128KDE Software suports up to four (4) emulated disk drives.

The 128KDE card has a short 16 conductor cable and connector which is used to access critical timing signals available only on the Apple mother board. This connector plugs into any one of the eight 4116 RAM sockets located at row "E" on the Apple mother board. See figure 1.1

The RAM IC which normally occupies this socket on the mother board is removed and installed in the vacant 4116 socket right above the ribbon cable on the 128KDE card and the ribbon cable is then plugged into the now vacant 4116 RAM socket on the Apple mother board.

* Apple Pascal and DOS 3.3 are trademarks of Apple Computer Inc

Installation (con't.)

Read these step-by-step instructions completely before attempting to install the 128KDE card in your Apple. Improper installation may result in damage to the 128KDE board and/or the Apple II computer. These instructions assume installation of the 128KDE card in slot Ø. If you have any doubts about installing the 128KDE card then ask your local Apple dealer for installation assistance.

1) Always turn off the power before removing or installing any peripheral cards in the Apple II computer. Remove the cover on the Apple by gently lifting the rear of the cover until the headlock fasteners unsnap.

2) Carefully remove the 4116 RAM IC at location E-3 on the Apple mother board with an IC removal tool. See figure 1.1

3) Place the 128KDE card in front of you, component side up, with the 16 conductor ribbon cable on the left side. Insert the 4116 RAM IC just removed from the Apple mother board into the empty socket just above the jumper cable on the 128KDE card. Make sure that the index notch on the RAM IC is to the left and pin one (1) is orientated with the number "1" on the 128KDE board. See figure 1.2

4) Insert the ribbon cable plug from the 128KDE card into the vacant socket at location E-3 on the Apple mother board with the ribbon cable on the left.

5) Install the 128KDE card into the slot Ø peripheral connector. Make sure that the card is seated all the way into the slot.

6) Place the cover back onto the Apple and press firmly until it snaps into the locked position.

That's all there is to it. You have just added an additional 128K of RAM memory to your Apple II computer and an almost "instant access" disk drive.

Insert the Legend 128KDE Demo disk into Drive 1 and boot-up in the normal manner as described in the Apple DOS 3.3 Manual. The disk operating system will be loaded into RAM memory and the HELLO program on the disk will automatically load Bank Ø on the 128KDE card with the language missing in ROM on the Apple mother board (Integer or Applesoft). Verify that the language has been loaded correctly by switching languages with the INT and FP commands. If the card is operating properly, you should be able to switch between languages.

If the Apple does not appear to be operating correctly then turn off the power immediately and check for proper installation of the 128KDE. If you are still having a problem contact your local Dealer. He should be able to help you isolate the problem.

Figure 1.1

Partial view of Apple motherboard

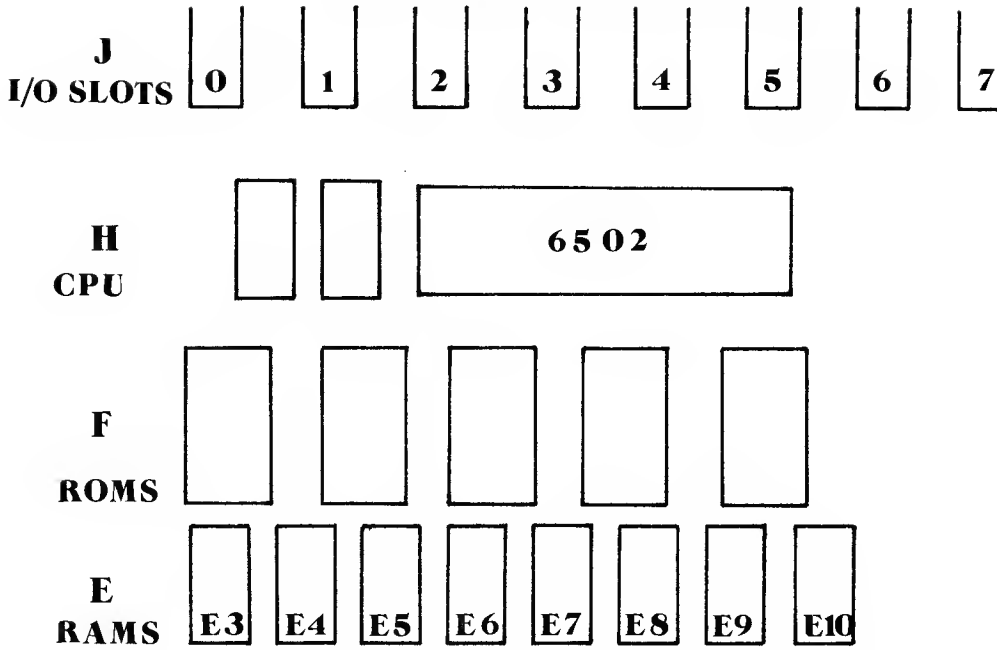
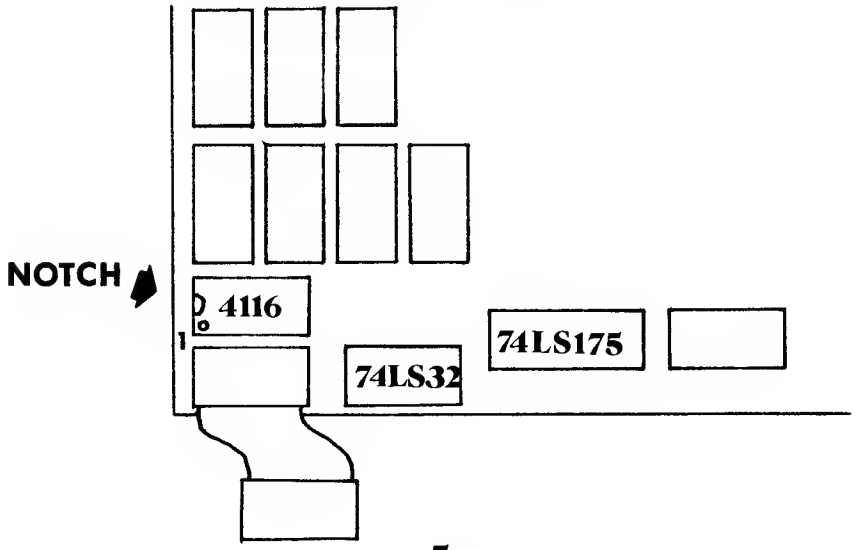


Figure 1.2

Orientation of 4116 on 128KDE

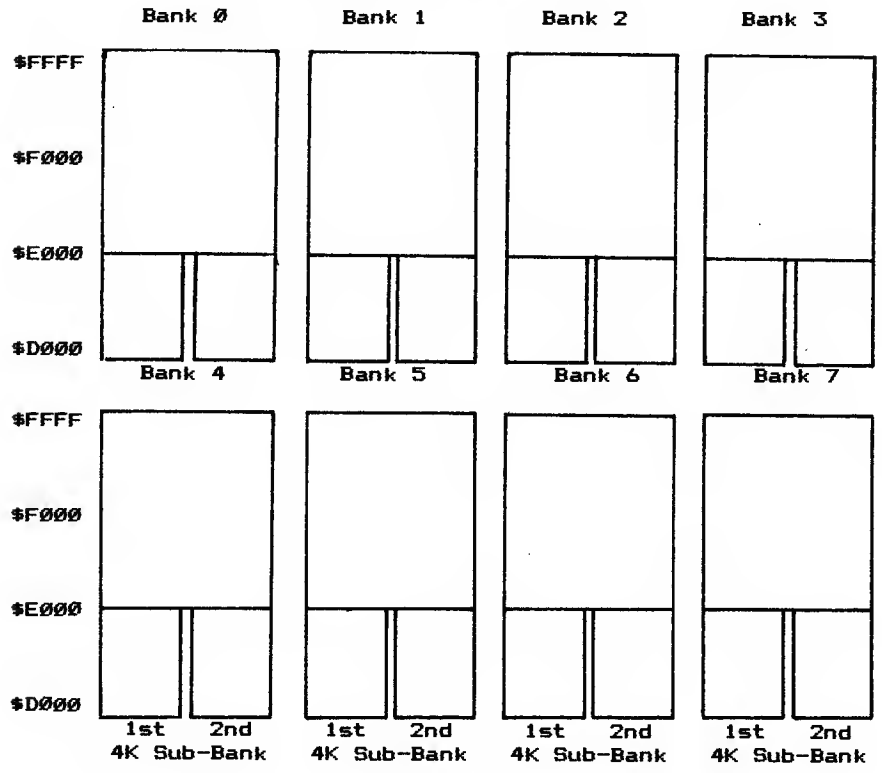


Theory of Operation

This chapter describes and details the operation of the 128KDE card at the machine and hardware level. It is not intended to serve as a tutorial but as a reference guide for the user. Use the Apple Reference Manual as an additional information source.

The RAM in the 128KDE card is configured as eight (8) individual 16K banks, numbered Ø through 7, which are addressed in the $DØØØ thru $FFFF address space in the Apple. This is the same address space used by the BASIC language contained in ROM (Read Only Memory) on the Apple mother board. The $DØØØ thru $FFFF address space represents only 12K of addressable memory and consequently each 16K bank of RAM is further divided into two 4K Sub-Banks mapped into the lower 4K address space from $DØØØ thru $DFFF. Please note that this is the same mapping convention used by the Apple Language Card and, in fact, the 128KDE card is functionally identical to eight (8) 16K Language Cards.

Figure 2.1
The 128KDE card Memory Map



Memory Management is the term used to describe the selection of one Bank of memory, either ROM or RAM, to be active in the same address space at any one time. The Apple II has used memory management for years to select the BASIC ROM's on the mother board or the BASIC ROM's on a firmware card in slot 0 to be mapped into the top 12K of address space in the Apple computer. This gives the user a choice of having either Integer or Applesoft BASIC active in the computer. The Apple Language Card is also mapped and managed in this same address space. The Language Card contains 16K of RAM memory that can be loaded with BASIC and behave like a BASIC ROM card or it can be loaded with Pascal or some other operating system making it much more flexible than a ROM card.

Bank 0 on the 128KDE card will always be recognized as a 16K Language Card when the card is installed in slot 0 and used with software designed to work with an Apple Language Card such as BASIC, Pascal, etc. Banks 1 through 7 will not be used and you may utilize each of these banks as you wish. If you 'boot' the Legend 128KDE System disk for example, it will load DOS 3.3 into the system in the 48K of RAM on the mother board and then the HELLO program will load Bank 0 on the 128KDE card with the language that is missing in ROM on the mother board. Similarly, Apple Pascal and other operating systems designed to work with a Language Card will utilize Bank 0 on the 128KDE card when it is installed in slot 0.

The important thing to remember about managing the memory in the 128KDE card is that only one bank of memory may be active in the $D000 thru $FFFF address space at any one time. When one of the 16K Banks of RAM on the 128KDE is selected and the card is "on" the Inhibit line (pin 32 on the Apple I/O connector) is pulled low and the ROM on the mother board is disabled. When the RAM is deselected on the 128KDE and the card is "off", the Inhibit line goes high and the ROM on the mother board is re-activated. If more than one 128KDE RAM card is installed in the Apple Computer, then care must be taken to select only one of the cards at any one time. If more than one card is selected, the Inhibit lines on the cards will conflict with each other and hang the system.

Selecting and deselecting the RAM on the 128KDE card is relatively easy. The 128KDE card is controlled by accessing one of the 16 Device Select addresses assigned to its slot. Each peripheral slot in the Apple has a unique set of 16 slot dependent Device Select addresses. These Device Select addresses were designed into the Apple for the purpose of controlling the hardware on a peripheral card.

Figure 2.2
Apple Device Select Addresses

Base Addr.	Slot 0	Slot 1	Slot 2	Slot 3	slot 4	Slot 5	Slot 6	Slot 7
$C080	$C080	$C090	$C0A0	$C0B0	$C0C0	$C0D0	$C0E0	$C0F0
$C081	$C081	$C091	$C0A1	$C0B1	$C0C1	$C0D1	$C0E1	$C0F1
$C082	$C082	$C092	$C0A2	$C0B2	$C0C2	$C0D2	$C0E2	$C0F2
$C083	$C083	$C093	$C0A3	$C0B3	$C0C3	$C0D3	$C0E3	$C0F3
$C084	$C084	$C094	$C0A4	$C0B4	$C0C4	$C0D4	$C0E4	$C0F4
$C085	$C085	$C095	$C0A5	$C0B5	$C0C5	$C0D5	$C0E5	$C0F5
$C086	$C086	$C096	$C0A6	$C0B6	$C0C6	$C0D6	$C0E6	$C0F6
$C087	$C087	$C097	$C0A7	$C0B7	$C0C7	$C0D7	$C0E7	$C0F7
$C088	$C088	$C098	$C0A8	$C0B8	$C0C8	$C0D8	$C0E8	$C0F8
$C089	$C089	$C099	$C0A9	$C0B9	$C0C9	$C0D9	$C0E9	$C0F9
$C08A	$C08A	$C09A	$C0AA	$C0BA	$C0CA	$C0DA	$C0EA	$C0FA
$C08B	$C08B	$C09B	$C0AB	$C0BB	$C0CB	$C0DB	$C0EB	$C0FB
$C08C	$C08C	$C09C	$C0AC	$C0BC	$C0CC	$C0DC	$C0EC	$C0FC
$C08D	$C08D	$C09D	$C0AD	$C0BD	$C0CD	$C0DD	$C0ED	$C0FD
$C08E	$C08E	$C09E	$C0AE	$C0BE	$C0CE	$C0DE	$C0EE	$C0FE
$C08F	$C08F	$C09F	$C0AF	$C0BF	$C0CF	$C0DF	$C0EF	$C0FF

The Device Select addresses used to control the LEGEND 128KDE card conform to the same Device Select address conventions used by Apple Computer, Inc. for implementing the Language Card with the following exceptions-

1) Apple uses only 8 of the 16 Device Select addresses available at any peripheral slot. Address bit 2 of the Device Select address is ignored on the Apple Language Card and therefore the addresses $C084-$C087 and $C08C-$C08F provide the same control functions as the $C080-$C083 and $C088-$C08B addresses. The LEGEND 128KDE responds to the $C080-$C084 and the $C088-$C08B Device Select addresses only.

2) The LEGEND 128KDE provides a Bank Select address at location $C084. This is a Write only address used to select bank 0 thru 7.

Any of the eight (8) banks can be selected by writing to the Bank Select Device Select address with a value from $00 to $07. The user can then use the other Device Select addresses to select ROM read, RAM read, Write enable RAM or Write protect RAM in either the first 4K bank or the second 4K bank. Refer to figure 2.3 for the device select address conventions used by the LEGEND 128KDE card (addresses used are for slot 0).

Use the information in Figure 2.2 (you may wish to refer to the Apple II Reference Manual, pg. 82) to substitute the correct Device Select addresses when accessing the 128KDE card from some slot other than 0. For example, the equivalent Device Select addresses in the $C0C0 through $C0CF range would be used for a 128KDE card installed in slot 4.

Theory of Operation (con't.)

For the novice and uninitiated, the following is a very simplistic procedure to follow. For the advanced Apple user, either disregard the following or you may find it interesting to read. In the following discussion, the Legend card is assumed to be installed in slot Ø. For other slot locations, please refer to Figure 2.2.

Step 1.

If you have a Disc Drive and/or an auto-start ROM, do not insert any disks in your drives at this time.

Step 2.

Turn on your Apple and your TV/Monitor. If an auto-start ROM is installed, press RESET to stop the disk drive.

Step 3.

A ">" symbol should now be in the upper left hand corner of the screen. If not, turn off the Apple and recheck Step 2.

Step 4.

Get into the monitor by typing 'Call-151'. The cursor should now be an asterisk "*". This is the cursor for the monitor.

Step 5.

At this point you have the option of doing several things.

a) You can select RAM read, and write protect RAM by typing CØ8Ø. However if you read RAM with no information in RAM you will 'hang' the system.

b) You can select ROM read and if you type the ROM read address $CØ81 two (2) times, you write enable the Legend card.

c) You can select ROM read and write protect the RAM, by typing CØ82. This will turn "off" the card.

d) You can select RAM read and if you type the RAM read address $CØ83 two (2) times you write enable the card with CØ83. However if you read RAM and there is no information in RAM you will 'hang' the system.

e) You can bank select by typing CØ84, a write only instruction to enable bank Ø through 7.

Write enable means you can write or put data into that (RAM) which is enabled.

Step 6.

First you must move information (the monitor routine) into the Legend card. To do this you must write enable the card and still be reading ROM. Type CØ81 <Press RETURN>, CØ81 <Press RETURN>. This is typed twice in order to write enable the card.

Theory of Operation (con't.)

Step 7.

You are now able to write information to the Legend card. Type the following to move the Apple monitor routine that is stored in ROM up into the Legend card: F8ØØ<F8ØØ:FFFFM. If this is done correctly you will return to the monitor cursor.

Step 8.

You now have a copy of the Apple monitor in the Legend card. For an example type in the following: 3ØØ:A9 C1 2Ø ED FD 18 69 Ø1 C9 DB DØ F6 6Ø <Press RETURN>. Now if you type 3ØØG, you should see the alphabet on the screen.

Step 9.

You may wish to select RAM read (now you can without hanging the system) and see what you have. You should find a copy of the Apple monitor routine around $Ø38Ø. You can do this by typing in Ø3ØØL, <Press RETURN> and listing (by typing the letter 'L') until you reach your destination.

Figure 2.3

128KDE Device Select Addresses for slot 0

2nd Bank	1st Bank	Function
$C080 -16256	$C088 -16248	Select RAM read and Write protect RAM
$C081 -16255	$C089 -16247	Select ROM read (2 reads will write enable RAM)
$C082 -16254	$C08A -16246	Select ROM read and Write protect RAM
$C083 -16253	$C08B -16245	Select RAM read (2 reads will write enable RAM)
$C084 -16252		Bank select (Write only to enable bank 0 through bank 7)

The following pages provide a few simple assembly language examples to help clarify the use of the Device Select addresses for selecting and deselecting the 128KDE card RAM. It should be pointed out that although there are many LDA instructions used in these examples the data in the accumulator is not used, it is the address itself that is sensed by the 128KDE card. You could just as easily use the LDX or other instructions as long as it references the Device Select address. The only exception is the Bank Select address ($C084) which is a 'Write only' address that requires that a value from $00 to $07 be written when the address is accessed. All of the following examples use the Device Select addresses for a 128KDE card installed in slot 0.

Example 1

This example shows how to deselect or turn off and write protect the RAM on the 128KDE card. Please note that the $C08A address could also have been used to perform this particular function.

```
0800            1  ;
0800            2  ; Deselect RAM read and write protect RAM.
0800            3  ; Select ROM read on the mother board.
0800            4  ;
0800            5  ;
0800 8D82C0     6  Start    LDA $C082         ;Deselect RAM rd.
0803 60         7           RTS               ;Return to caller
0804            8  ;
                9           END
```


Example 2

This example shows how to deselect or turn off the 128KDE card RAM read and enable ROM read on the mother board. This example routine also write enables Bank 5 on the 128KDE card with the 1st 4K sub-bank mapped into the $D000-$DFFF address space. While we have the ROM read enabled we are able to write data into RAM on the 128KDE card.

```
0800            1  ;
0800            2  ; Select ROM read & Write enable 1st 4k sub-
0800            3  ; bank of Bank #5 with 2 consecutive reads.
0800            4  ;
0800            5  BankSel  EQU $C084
0800            6  ;
0800 A905       7  Start    LDA #$05          ;Data for Bank #5
0802 8D84C0     8           STA BankSel       ;Select Bank no.5
0805 AD89C0     9           LDA $C089         ;Select ROM read.
0808 AD89C0    10           LDA $C089         ;Write enable RAM
080B 60        11           RTS               ;Return to caller
080C           12  ;
               13           END
```

A copy of the Monitor could be installed in any one of the eight (8) 16K banks in the 128KDE card using the following routine (example 3). Simply change the operand in line 7 from #$00 to #$01, #$02 etc. through #$07 to select Bank 1, Bank 2 through Bank 7 respectively.

Example 3

This example shows how to deselect or turn off the 128KDE card RAM read and enable ROM read on the mother board. This example also write enables Bank Ø on the 128KDE card with the 2nd 4K sub-bank mapped into the $DØØØ-$DFFF address space. While we have the ROM read enabled we are able to write data into RAM on the 128KDE card. This example is expanded to show a very short routine called COPYMON which copies the Apple Monitor ROM on the mother board at address $FBØØ to $FFFF into the same address space in Bank Ø on the 128KDE card.

Notice that the routine uses only one set of zero page pointers, PTRL and PTRH. Of particular importance here is the fact that any load instruction (line 2Ø) will read ROM on the mother board while any store instruction (line 21) will write to the 128KDE RAM at the same address contained in PTRL and PTRH.

```
ØBØØ             1  ;
ØBØØ             2  ; Select ROM read & Write enable 2nd 4k sub-
ØBØØ             3  ; bank of Bank #Ø with 2 consecutive reads.
ØBØØ             4  ;
ØBØØ             5  BankSel  EQU $CØ84
ØBØØ             6  ;
ØBØØ A9ØØ       7  Start    LDA ##$ØØ          ;Data for Bank #Ø
ØBØ2 8D84CØ      8           STA BankSel        ;Select Bank no.Ø
ØBØ5 AD81CØ      9           LDA $CØ81          ;Select ROM read.
ØBØ8 AD81CØ     1Ø           LDA $CØ81          ;Write enable RAM
ØBØB            11  ;
ØBØB            12  PTRL     EPZ $Ø6            ;Zero Pg variable
ØBØB            13  PTRH     EPZ $Ø7            ;Zero Pg variable
ØBØB            14  ;
ØBØB A9ØØ       15  COPYMON  LDA ##$ØØ          ;Address lo byte.
ØBØD 85Ø6       16           STA PTRL           ;Pointer lo byte.
ØBØF A9F8       17           LDA ##$F8          ;Address hi byte.
Ø811 85Ø7       18           STA PTRH           ;Points to $FBØØ.
Ø813 AØØØ       19           LDY ##$ØØ          ;Set Y index to Ø
Ø815 B1Ø6       2Ø  CopyROM  LDA (PTRL),Y       ;Get byte fm ROM.
Ø817 91Ø6       21           STA (PTRL),Y       ;Put byte in RAM.
Ø819 C8         22           INY                ;All 256 bytes  ?
Ø81A DØF9       23           BNE CopyROM        ;No, copy another
Ø81C E6Ø7       24           INC PTRH           ;Copied 8 pages ?
Ø81E DØF5       25           BNE CopyROM        ;No, copy another
Ø82Ø            26  ;
Ø82Ø            27  ; Write protect RAM and return to caller.
Ø82Ø            28  ;
Ø82Ø AD82CØ     29           LDA $CØ82          ;Select ROM read.
Ø823 6Ø         3Ø           RTS                ;Return to caller
Ø824            31  ;
                32           END
```


Example 4

One of the interesting aspects of having a language contained in RAM on the 128KDE card is that it can be modified easily, something you can't do with Read Only Memory (ROM). It just so happens that a copy of the Apple Auto-Start Monitor resides at address $F800 to $FFFF in Bank 0 on the card when it is loaded with a language by the DOS 3.3 System Master Disk (the Auto-Start Monitor is listed in the Apple II Reference Manual, pg.136). Many users modify the Auto-Start Monitor to include custom functions. These custom Monitors can be copied into the RAM on the 128KDE card and utilized when the user switches to the language contained in RAM on the card (Integer or Applesoft).

This routine assumes that you have just BLOADed your own custom Monitor into memory at address $1000 and that Bank 0 on the 128KDE card has already been loaded with a language. This routine will copy the custom Monitor into Bank 0 RAM.

```
0800             1  ;
0800             2  ; Select ROM read & Write enable 2nd 4k sub-
0800             3  ; bank of Bank #0 with 2 consecutive reads.
0800             4  ;
0800             5  BankSel  EQU $C084
0800             6  ;
0800 A900        7  Start    LDA #$00           ;Data for Bank #0
0802 8D84C0      8           STA BankSel        ;Select Bank no.0
0805 AD81C0      9           LDA $C081          ;Select ROM read.
0808 AD81C0     10           LDA $C081          ;Write enable RAM
080B            11  ;
080B            12  PTRL     EPZ $06            ;Zero Pg variable
080B            13  ADRL     EPZ $08            ;Zero Pg variable
080B            14  ;
080B A900       15  COPYMON  LDA #$00           ;Address lo byte.
080D 8506       16           STA PTRL           ;Pointer lo byte.
080F 8508       17           STA ADRL           ;Pointer lo byte.
0811 A9F8       18           LDA ##F8           ;Address hi byte.
0813 8507       19           STA PTRL+1         ;Points to $F800.
0815 A910       20           LDA #$10           ;Address hi byte.
0817 8509       21           STA ADRL+1         ;Points to $1000.
0819 A000       22           LDY #$00           ;Set Y index to 0
081B B108       23  CopyROM  LDA (ADRL),Y       ;Get source byte.
081D 9106       24           STA (PTRL),Y       ;Put byte in RAM.
081F C8         25           INY                ;All 256 bytes  ?
0820 D0F9       26           BNE CopyROM        ;No, copy another
0822 E609       27           INC ADRL+1         ;Inc page pointer
0824 E607       28           INC PTRL+1         ;Copied 8 pages ?
0826 D0F3       29           BNE CopyROM        ;No, copy another
0828            30  ;
0828            31  ; Write protect RAM and return to caller.
0828            32  ;
0828 AD82C0     33           LDA $C082          ;Select ROM read.
082B 60         34           RTS                ;Return to caller
082B            35  ;
                36           END
```

Memory Master

Memory Master is a memory management program on the 128KDE system disk. Memory Master is used in conjunction with the Apple II computer equipped with DOS 3.3 and a 128KDE card. This program will work with most 16K cards currently available and provides a full 44K bytes of program storage within the 48K on the Apple 11 motherboard. With the use of this program, an additional 8.5K of RAM is made available by relocating DOS into one of the first four (4) 16K banks on the 128KDE card.

Memory Master will also manage your Integer or Applesoft card, providing you have one installed, in any slot in your Apple 11.

Memory Master provides many features and several powerful extensions to the standard Apple 1I DOS 3.3 Disk Operating System.

1) Machine language programs can access the DOS RWTS routines through the standard DOS page 3 vectors ($3D0 through $3EC) and no additional page 3 space is used by MEMORY MASTER.

2) The <.FL1P> command allows the user to "flip" between DOS 3.3 and DOS 3.2 without re-booting, and best of all, any programs residing in memory are unaffected by the "flip". What a great way to move a single file from DOS 3.3 to DOS 3.2, or vice-versa.

3) The <.SHOW> command promptly displays the current DOS version in use.

4) The <.BSTAT> command dislpays the hexidecimal starting address and length of the last Binary file either BLOADed or BRUN, a valuable feature for BSAVEing Binary files.

5) The <.MON1TOR> command places the user into the Apple Monitor.

Memory Master is very flexible. Almost any combination of cards can be used in the system. Here are some possible system configuration considerations.

1) If you are not using a F1RMWARE card then you should install the 128KDE in slot 0. Bank 0 on the 128KDE card will be loaded with the missing BAS1C language when booting the system diskette. In addition, the Pascal Operating System will use the card when installed in slot 0.

2) If you are using your F1RMWARE card then you should install it in slot 0 and put the 128KDE in one of the other slots (any one available will do).

Insert the 128KDE System diskette into the drive and boot-up in the normal manner as described in the Apple DOS Manual and then type-

```
>BRUN MEMORY MASTER 2.0  <Press RETURN>
```

The program will begin execution after it has been loaded into memory and checks to make sure that DOS is already in the machine at the correct address. If there is any problem the program returns to BASIC and displays the following message-

```
INCORRECT DOS INSTALLED IN THE MACHINE.
```

Set-up

The program will prompt the user with three important set-up questions. These questions are-

```
WHAT SLOT IS FIRMWARE CARD IN (0-7) ? 0
WHAT SLOT IS THE RAM  CARD IN (0-7) ? 0
WHICH BANK SHOULD DOS GO INTO (0-3) ? 0
```

The user supplied answers to these questions are used to set up the memory management routines incorporated into MEMORY MASTER. Let's go through these three questions, one at a time.

```
WHAT SLOT IS FIRMWARE CARD IN (0-7) ? 0
```

Type in the slot number, in the range from 0 through 7, of the Integer or Applesoft FIRMWARE card. If no FIRMWARE card is installed then simply press RETURN.

```
WHAT SLOT IS THE RAM  CARD IN (0-7) ? 0
```

Type in the slot number, in the range from 0 through 7, of the LEGEND 128KDE card (or 16K RAM card if you're not using the LEGEND 128KDE card).

```
WHICH BANK SHOULD DOS GO INTO (0-3) ? 0
```

If you are using one of the 16K RAM cards then simply press RETURN in response to this question.

If you are using the LEGEND 128KDE card then type in the Bank number, in the range from 0 through 3, to be used for the relocated Disk Operating System. Please note, when the 128KDE card is installed in slot 0 that Bank 0 on the card will contain the Language missing in ROM on the Apple mother board. Use Bank number 1, 2 or 3 for the relocated DOS. If a FIRMWARE card is being used (question 1) then you may use any of the four Banks on the 128KDE card for DOS.

The modified Disk Operating System installed in RAM on the LEGEND 128KDE card (or 16K RAM card) by MEMORY MASTER is used in exactly the same manner as standard Apple DOS 3.3, with the following exceptions-

1) The DOS "INIT" command has been disabled. When you need to initialize diskettes simply boot-up the SYSTEM MASTER diskette which loads standard (and unmodified) Apple DOS into the machine and then initialize your blank diskettes.

2) The DOS "CATALOG" command displays the number of unused sectors remaining on each diskette as you catalog it.

3) Four new DOS commands have been added to the system. These new commands and their uses are described below.

.F or .FLIPDOS

The ".FLIP" command will automatically switch between DOS 3.3 and DOS 3.2. The command works in either direction, that is, if you are currently in DOS 3.3 then the flip command will place you into DOS 3.2, and vice-versa.

The ".FLIP" command can be typed in directly from the Apple keyboard or it can be used inside your programs by using the standard Apple DOS print control-D syntax. Example-

```
>100 D$="":REM CTRL-D
>110 PRINT D$ ; ".FLIP"
```

All programs, pointers and variables remain unchanged during the "FLIP" operation. This gives the user the capability to transfer files between different types of DOS diskettes (13 and 16 sector) with a simple "LOAD", ".FLIP" and "SAVE" sequence of operations.

.S or .SHOWDOS

The ".SHOWDOS" command, when typed in from the keyboard, displays the current DOS in use, either DOS 3.3 (16 sector) or DOS 3.2 (13 sector).

.B or .BSTAT

The ".BSTAT" command, when typed in from the keyboard, displays the hexidecimal starting address and length of the last Binary type file that was either BLOADed or BRUN.

This command provides the user with the starting address and length parameters required by Apple DOS when BSAVEing Binary type files onto disk.

Memory Master (con't.)

.M or .MONITOR

The ".MON" command, when typed in from the keyboard, places the user into the Apple Monitor.

Memory Usage

Normally Apple DOS 3.3 will reside from $9600 through $BFFF in a 48K Apple and DOS will set HIMEM to a value of $9600 (decimal 38400 or -27136). MEMORY MASTER uses the top eight pages of memory on the mother board for the Memory Management Routines ($BF00-$BFFF) and the three DOS File Buffers ($B800-$BEFF) and the rest of DOS is relocated into RAM on the LEGEND 128KDE card. This allows DOS to set HIMEM to a value of $B800 (decimal 47104 or -18432).

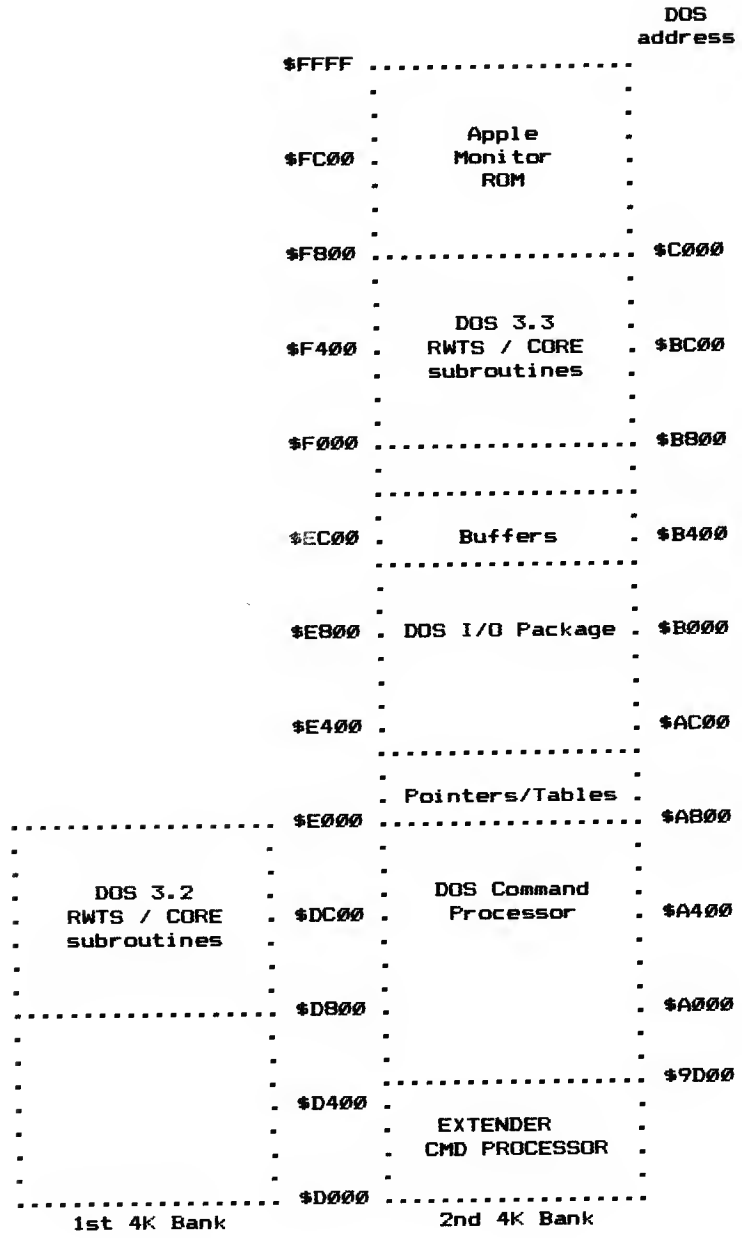
RWTS Access

MEMORY MASTER provides machine language access to the DOS Read Write Track & Sector (RWTS) routines through the standard DOS page 3 vectors as described in the Apple DOS Manual, Chapter 9, pages 94 - 98. The DOS IOB is not relocated into RAM on the LEGEND 128KDE card but is moved to the $BFE8-$BFF8 address range within the Memory Management Routines for direct access by your machine language programs.

Limitations

MEMORY MASTER remains active in memory until the user re-boots. It should be pointed out that any program disks that must be booted into operation will disconnect MEMORY MASTER and install their own DOS into memory at the standard 48K address ($9600-$BFFF) in the machine. These programs, unfortunately, can not take advantage of the extra memory and other features of the MEMORY MASTER system. Also, any programs that modify pointers internal to DOS or access RWTS directly and not through the page 3 vectors, will not work correctly with the MEMORY MASTER system active.

Memory Master Memory Map



```
24D5           609  ********************************************
24D5           610  *                                          *
24D5           611  * <DOS MEMORY MANAGEMENT ROUTINES> by Mac  *
24D5           612  *(C) Copyright 1981-Legend Industries, Ltd.*
24D5           613  ;
BF00           614           ORG $BF00
BF00           615           OBJ BLOCK1-$1800
BF00           616  ;
BF00 203FBF    617  DOSWARM  JSR DOSON          ;.................
BF03 4CBFD5    618           JMP $9DBF+$3800    ;.................
BF06 203FBF    619  DOSCOLD  JSR DOSON          ;.................
BF09 4C84D5    620           JMP $9D84+$3800    ;.................
BF0C 203FBF    621  IOPKG    JSR DOSON          ;.................
BF0F 20FDE2    622           JSR $AAFD+$3800    ;.................
BF12 4C63BF    623           JMP DOSOFF         ;.................
BF15 203FBF    624  GORWTS   JSR DOSON          ;.................
BF18 20B5EF    625           JSR $B7B5+$3800    ;.................
BF1B 4C63BF    626           JMP DOSOFF         ;.................
BF1E 203FBF    627  PKGLOC   JSR DOSON          ;.................
BF21 AD0FD5    628           LDA $9D0F+$3800    ;.................
BF24 AC0ED5    629           LDY $9D0E+$3800    ;.................
BF27 4C63BF    630           JMP DOSOFF         ;.................
BF2A 203FBF    631  IOBLOC   JSR DOSON          ;.................
BF2D ADC2E2    632           LDA $AAC2+$3800    ;.................
BF30 ACC1E2    633           LDY $AAC1+$3800    ;.................
BF33 4C63BF    634           JMP DOSOFF         ;.................
BF36 203FBF    635  DOSHOOKS JSR DOSON          ;.................
BF39 2051E0    636           JSR $A851+$3800    ;.................
BF3C 4C63BF    637           JMP DOSOFF         ;.................
BF3F           638  ;
BF3F 48        639  DOSON    PHA                ;Save accumulator
BF40 AD82C0    640           LDA $C082          ;Write prot.cd.#0
BF43 AD81C0    641           LDA $C081          ;Turn off slot #0
BF46 ADFABF    642           LDA BANKNBR        ;.................
BF49 8D84C0    643           STA $C084          ;Select DOS bank#
BF4C AD83C0    644           LDA $C083          ;Turn on slot # X
BF4F AD83C0    645           LDA $C083          ;Write enable # X
BF52 68        646           PLA                ;.................
BF53 60        647           RTS                ;.................
BF54           648  ;
BF54 203FBF    649  DOSIN    JSR DOSON          ;Select DOS bank.
BF57 2081D6    650           JSR $D681          ;DOS input rout's
BF5A 4C63BF    651           JMP DOSOFF         ;.................
BF5D 203FBF    652  DOSOUT   JSR DOSON          ;Select DOS bank.
BF60 20BDD6    653           JSR $D6BD          ;DOS Output routs
BF63           654  ;
BF63 08        655  DOSOFF   PHP                ;Save P.
BF64 48        656           PHA                ;Save A.
BF65 A900      657           LDA #$00           ;.................
BF67 8D84C0    658           STA $C084          ;.................
BF6A AD82C0    659           LDA $C082          ;Turn off slot #X
BF6D ADF9BF    660           LDA LANGID         ;Get lang.ID byte
```



```
BF70 CD00E0    661           CMP $E000          ;Sameas mtherROM?
BF73 F003      662           BEQ EXIT1          ;Yes,ret.2 caller
BF75 AD80C0    663           LDA $C080          ;Turn on slot # 0
BF78 68        664  EXIT1    PLA                ;Restore A.
BF79 28        665           PLP                ;Restore P.
BF7A 60        666           RTS                ;.................
BF7B 2063BF    667  ROMSW    JSR DOSOFF         ;.................
BF7E 2C81C0    668           BIT $C081          ;Turn off slot #0
BF81 CD00E0    669           CMP $E000          ;Is lang.in ROM ?
BF84 F00E      670           BEQ SETLANG        ;.................
BF86 2C80C0    671           BIT $C080          ;Is lang.in RAM ?
BF89 CD00E0    672           CMP $E000          ;.................
BF8C F006      673           BEQ SETLANG        ;.................
BF8E 08        674  EXIT2    PHP                ;Save the Z flag.
BF8F 203FBF    675           JSR DOSON          ;.................
BF92 28        676           PLP                ;.................
BF93 60        677           RTS                ;.................
BF94 8DF9BF    678  SETLANG  STA LANGID          ;Save lang.ID byt
BF97 F0F5      679           BEQ EXIT2          ;.................
BF99           680  ;
BF99 2063BF    681  I.CHAIN  JSR DOSOFF         ;.................
BF9C 4C36E8    682           JMP $E836          ;.................
BF9F 2063BF    683  I.ERR    JSR DOSOFF         ;.................
BFA2 4CE3E3    684           JMP $E3E3          ;.................
BFA5 2063BF    685  I.COLD   JSR DOSOFF         ;.................
BFA8 4C00E0    686           JMP $E000          ;.................
BFAB 2063BF    687  I.WARM   JSR DOSOFF         ;.................
BFAE 4C03E0    688           JMP $E003          ;.................
BFB1 2036BF    689  A.CHAIN  JSR DOSHOOKS       ;.................
BFB4 2065D6    690           JSR $D665          ;.................
BFB7 8533      691           STA $33            ;.................
BFB9 85D8      692           STA $D8            ;.................
BFBB 4CD2D7    693           JMP $D7D2          ;.................
BFBE 2063BF    694  A.ERR    JSR DOSOFF         ;.................
BFC1 4C65D8    695           JMP $D865          ;.................
BFC4 2063BF    696  A.WARM   JSR DOSOFF         ;.................
BFC7 4C3CD4    697           JMP $D43C          ;.................
BFCA 2063BF    698  A.RELO   JSR DOSOFF         ;.................
BFCD 20F2D4    699           JSR $D4F2          ;.................
BFD0 4C3FBF    700           JMP DOSON          ;.................
BFD3           701  ;
BFD3 6C3600    702  HOOKCSW  JMP ($0036)
BFD6 6C3800    703  HOOKKSW  JMP ($0038)
BFD9           704  ;
BFD9 000000    705           HEX 000000         ;.................
BFDC 000000    706           HEX 000000         ;.................
BFDF 000000    707           HEX 000000         ;.................
BFE2 000000    708           HEX 000000         ;.................
BFE5 000000    709           HEX 000000         ;.................
BFE8           710  ;
```

```
BFE8 Ø1       711  1BYTE    HEX Ø1             ;.................
BFE9 6Ø       712  1BSLOT   HEX 6Ø             ;.................
BFEA Ø1       713  1BDRVN   HEX Ø1             ;.................
BFEB ØØ       714  1BVOL    HEX ØØ             ;.................
BFEC 11       715  1BTRK    HEX 11             ;.................
BFED ØØ       716  1BSECT   HEX ØØ             ;.................
BFEE FBB7     717  1BDCTP   ADR $B7FB          ;.................
BFFØ E8B7     718  1BBUFP   ADR $B7E8          ;.................
BFF2 ØØØØ     719          HEX ØØØØ           ;.................
BFF4 Ø1       72Ø  1BCMD    HEX Ø1             ;.................
BFF5 ØØ       721  1BSTAT   HEX ØØ             ;.................
BFF6 FE       722  1BSMOD   HEX FE             ;.................
BFF7 6Ø       723  1OBPSN   HEX 6Ø             ;.................
BFF8 Ø1       724  1OBPDN   HEX Ø1             ;.................
BFF9          725  ;
BFF9 ØØ       726  LANGID   HEX ØØ             ;.................
BFFA ØØ       727  BANKNBR  HEX ØØ             ;.................
BFFB          728  ;
BFFB ØØ       729  DEVPTC   HEX ØØ             ;.................
BFFC Ø1       73Ø  PPTC     HEX Ø1             ;.................
BFFD EFD8     731  MONTC    HEX  EFD8          ;.................
BFFF          732  ;
BFFF B3       733  DOSTYPE  HEX B3             ;DOS version 3.3
CØØØ          734  *                                             *
CØØØ          735  ************************************************
CØØØ          736  ;
CØØØ          737          END
```

**** END OF ASSEMBLY

Disk Emulator

The DISK EMULATOR is a remarkably efficient Memory Management program for the Apple II computer. This program is on the 128KDE System diskette and simulates up to four additional, almost instant access, disk drive units for the storage and retreival of standard DOS 3.3 disk files.

Now, for the first time, every byte of RAM memory on the LEGEND 128KDE card can be accessed with the standard DOS 3.3 disk commands ie. LOAD, SAVE, OPEN, READ, WRITE, etc. in BASIC, or the DOS RWTS subroutines in machine language.

DISK EMULATOR is remarkably compact, using only 512 bytes of memory when installed within DOS, yet so very powerful because it takes advantage of the organization and power built into the DOS 3.3 Disk Operating System.

A pseudo SLOT/DRIVE number is used to access the emulated drive. The user can assign any SLOT/DRIVE number to the DISK EMULATOR, and the EMULATOR will not interfere with the operation of a peripheral card installed in that slot inside the Apple.

DISK EMULATOR is as much as 300% faster than the Apple Disk II drive because it eliminates the motor speed, step, nybble and search delays associated with the Disk II hardware.

DISK EMULATOR will use a single 128KDE card to simulate 512 sectors (tracks 3 thru 34). Tracks Ø,1, and 2 are reserved for DOS.

DISK EMULATOR will support up to four Legend 128KDE cards providing a total of 512K bytes (half megabyte) of online memory, all organized as four 128K byte, fast access, disk drives.

DISK EMULATOR provides a new <Mount> command, which quickly copies the contents of a floppy diskette into the specified emulated disk, and a new <Update> command, which copies the contents of the specified emulated disk back onto floppy diskette, within 18 seconds.

DISK EMULATOR, like all of the Memory Management software from Legend Industries, will support your Integer or Applesoft FIRMWARE card (if you have one) installed in any slot.

The DISK EMULATOR system supports several different configurations. If the Legend 128KDE RAM card is installed in slot Ø and an Integer or Applesoft FIRMWARE card is installed in some other slot then DISK EMULATOR will utilize all of the RAM on the 128KDE to provide 512 sectors (128K bytes) of storage and use the FIRMWARE card to provide both BASIC languages.

Disk Emulator (con't.)

If the Legend 128KDE RAM card is installed in slot 0 and NO Integer or Applesoft FIRMWARE card is installed, then DISK EMULATOR will retain the first 16K Bank on the card (Bank 0) for the language missing in ROM on the mother board and utilize Banks 1 through 7 on the 128KDE to provide 448 sectors (112K bytes) of storage.

If a 16K RAM card, like the Apple Language card, is installed in slot 0 and the Legend 128KDE card is installed in some other slot then DISK EMULATOR will utilize all eight (8) 16K banks of RAM on the 128KDE to provide 512 sectors (128K bytes) of storage and the 16K RAM card in slot 0 will be used for the language missing in ROM on the Apple mother board.

The DISK EMULATOR program needs to know the location of the FIRMWARE card, if any, the Emulated slot and drive number(s) and the location of the Legend 128KDE card(s) in the Apple before it installs itself into the DOS 3.3 Disk Operating System. The user can supply this information <Manually> by BRUNing the DISK EMULATOR program, or the user can supply this information with a <Turnkey> Applesoft program to 'bring-up' the DISK EMULATOR system automatically.

Insert the 128KDE System diskette into the drive and boot-up in the normal manner as described in the Apple DOS manual and then type-

>BRUN DISK EMULATOR 4.0 <Press RETURN>

The program will begin execution after it has been loaded into memory and checks to make sure that DOS is already in the machine at the correct address. If there is any problem, the program returns to BASIC and displays the following message-

INCORRECT DOS INSTALLED IN THE MACHINE.

If there are no problems, the Apple screen display should look something like this-

```
( DISK EMULATOR 4.0 )(C) COPYRIGHT, 1981
(MICHAEL D. MC LAREN) LEGEND INDUSTRIES

  ROM FIRMWARE CARD ( LOCATION- SLOT 0 )

  DISK EMULATOR #1  ( SLOT X / DRIVE X )
                    ( 128K CARD SLOT X )

  DISK EMULATOR #2  ( SLOT X / DRIVE X )
                    ( 128K CARD SLOT X )

  DISK EMULATOR #3  ( SLOT X / DRIVE X )
                    ( 128K CARD SLOT X )

  DISK EMULATOR #4  ( SLOT X / DRIVE X )
                    ( 128K CARD SLOT X )
```


Disk Emulator (con't.)

Let's input the parameters required by DISK EMULATOR one at a time.

*** ROM FIRMWARE CARD (LOCATION- SLOT Ø)**

Type in the slot number, in the range from Ø through 7, of the Integer or Applesoft FIRMWARE card. If no FIRMWARE card is installed then simply press <RETURN>.

The DISK EMULATOR system will manage the selection of the language contained on the FIRMWARE card or the language contained in ROM on the mother board when the location of the FIRMWARE card is reported here.

The DISK EMULATOR system will use slot Ø as the default slot for the language missing in ROM on the mother board when no FIRMWARE card is present and consequently will recognize a 16K RAM card like the Apple Language card or Bank Ø on the Legend 128KDE RAM card when either of these cards is present in slot Ø.

*** DISK EMULATOR #1 (SLOT X / DRIVE X)**

Type in the slot number first, in the range from Ø through 7, and then the drive number, either 1 or 2, that the first Disk Emulator will respond to. If you have a single Disk II drive on your Apple II at Slot 6 / Drive 1 then use SLOT 6 / DRIVE 2 for the emulated drive. If you have two Disk II drives at Slot 6 / Drives 1 and 2 then use SLOT 5 / DRIVE 1 for the emulated drive.

You can use any slot for the emulated drive and it will not interfere with the operation of a peripheral card installed in that slot. This means that you can use SLOT 5 / DRIVE 1 for the emulated drive and it will not interfere with the operation of a printer or a modem or any type of card that is physically installed in that slot in the Apple.

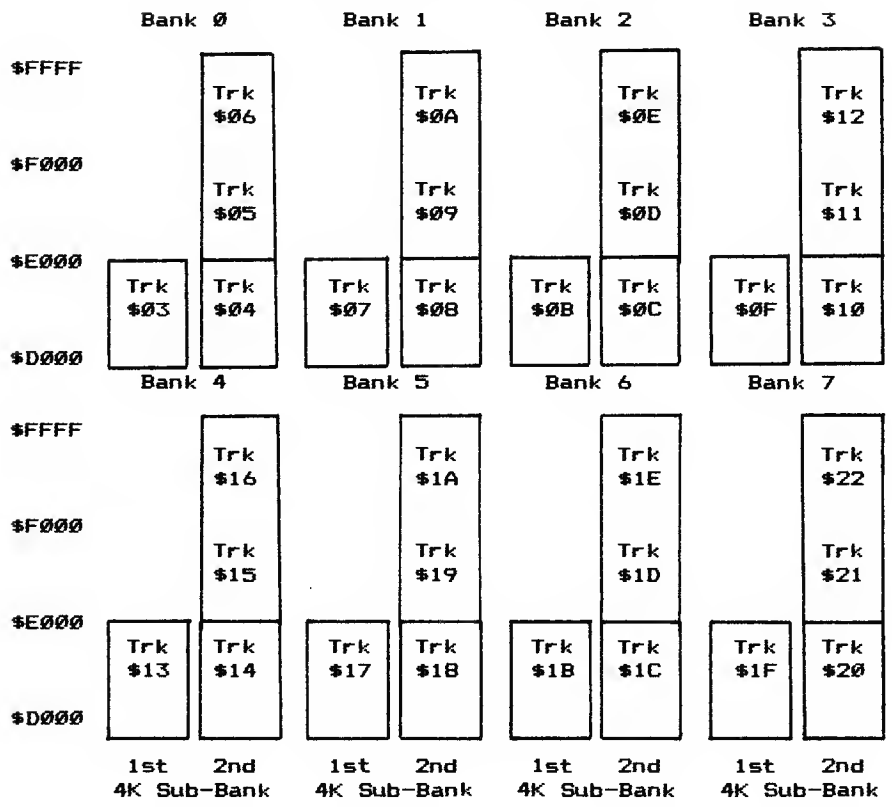
******** Special Note ******** You should not assign a SLOT / DRIVE number to the emulator that is already being used by an Apple Disk II drive because this will disable the Disk II drive.

*** (128K CARD SLOT X)**

Type in the slot number, in the range from Ø through 7, of the Legend 128KDE RAM card.

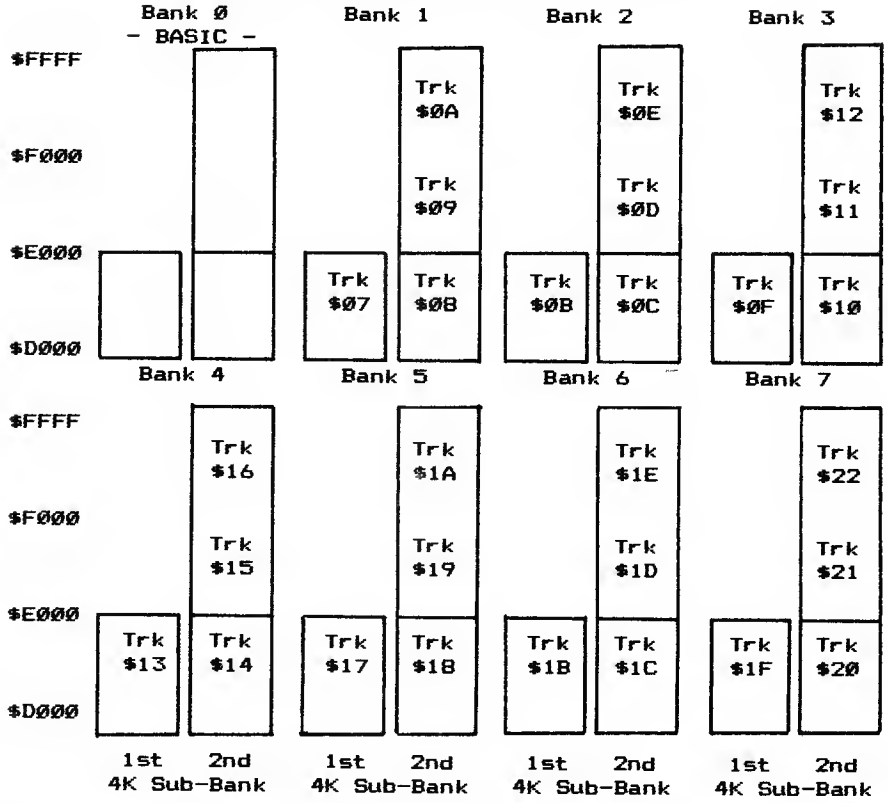
If a FIRMWARE card is present, or if the Legend 128KDE card is installed in any slot other than slot Ø, then DISK EMULATOR will use this card for the storage of 512 sectors of information (128K bytes) organized as disk tracks 3 through 34 as shown in Figure 3.1.

Figure 3.1
128KDE card Memory Map without BASIC



If a FIRMWARE card is not present and the Legend 128KDE card is installed in slot Ø then DISK EMULATOR will retain the first 16K bank of RAM on the card (Bank Ø) for the language missing in ROM on the Apple mother board and will utilize Banks 1 through 7 for the storage of 448 sectors of information (112K bytes) organized as disk tracks 7 through 34 as shown in Figure 3.2.

Figure 3.2
128KDE card Memory Map with BASIC



DISK EMULATOR does not provide for the storage of tracks Ø, 1 and 2 on the emulated drive. These tracks usually contain the Disk Operating System (DOS) on a floppy disk and are not normally available for the storage of disk files.

You have just supplied the parameters necessary for operation of EMULATOR #1. If you are not going to implement EMULATOR #2, EMULATOR #3 or EMULATOR #4 then simply press <RETURN> in response to the emulated Slot / Drive prompts and the 128K card Slot assignment prompts for each of the remaining three emulators.

If you have more than two Legend 128KDE cards and you wish to implement EMULATOR #2 then type in the Slot and Drive that this emulator will respond to as well as the location of the second Legend 128KDE cards just like you did for EMULATOR #1. Similarly, type in the parameters for EMULATOR #3 and EMULATOR #4 if you wish, or simply press <RETURN> in response to the emulated Slot / Drive and 128K Slot prompts for the last two EMULATORS.

* (A)BORT, (I)NSTALL OR (R)ESTORE ? I

Press 'A' to Abort and return to BASIC without installing or restoring the DISK EMULATOR system.

Press 'I' to Install the DISK EMULATOR system and initialize the Directory on all of the emulated disks. DISK EMULATOR will automatically set up the Directory on each emulated disk with 496 free sectors (432 free sectors if Bank Ø on the 1st 128KDE card contains BASIC).

Press 'R' to Restore or reconnect the DISK EMULATOR system without initializing the Directory on the Emulated disks. DISK EMULATOR is disconnected when you re-boot DOS into the Apple, the Restore command is used to reconnect the DISK EMULATOR with all of the information on the emulated disks intact.

The Legend Industries DISK EMULATOR system can be installed into the Apple with an Applesoft <Turnkey> program, thus eliminating the need to type in all of the Slot and Drive parameters that DISK EMULATOR requires for operation. The user can RUN the <Turnkey> program to install DISK EMULATOR and then display the CATALOG on a disk or the <Turnkey> program can be used as the greeting program on a disk to automatically install DISK EMULATOR and then RUN some other BASIC program without user intervention.

The <Turnkey> program will BLOAD the DISK EMULATOR file into memory and then POKE in the required Slot and Drive parameters before installing DISK EMULATOR into the Apple. The <Turnkey> program remains in control of the Apple after the installation and consequently the last line in the program may be a disk command such as CATALOG, to display the Directory of the disk, or RUN, to run the users program.

Before you can use the <Turnkey> program you must customize it to reflect the Slot and Drive numbers of the emulated disks and the location of the Legend 128KDE RAM cards installed in your Apple. The sample <Turnkey> program called 'TURNKEY' on your LEGEND disk is listed for your convenience in Listing 1. This program can be modified to reflect the configuration of your DISK EMULATOR system.

LOAD the TURNKEY program into memory and retype line 1ØØ through line 24Ø using the following guidelines on the next page.

100 RC = 0
Use a value in the range from 0 through 7 that corresponds to the location of the Integer or Applesoft FIRMWARE card. If no FIRMWARE card is installed then use the value 0.

110 S1 = 6
Use a value in the range from 0 through 7 that corresponds to the SLOT number that EMULATOR #1 will respond to.

120 D1 = 2
Use the value 1 or 2 that corresponds to the DRIVE number that EMULATOR #1 will respond to.

130 K1 = 0
Use a value in the range from 0 through 7 that corresponds to the location of the 128KDE card.

160 S2 = 15
Use a value in the range from 0 through 7 that corresponds to the SLOT number that EMULATOR #2 will respond to or use the value 15 to indicate that EMULATOR #2 will not be implemented.

170 D2 = 15
Use the value 1 or 2 that corresponds to the DRIVE number that EMULATOR #2 will respond to or use the value 15 to indicated that EMULATOR #2 will not be implemented.

180 K2 = 15
Use a value in the range from 0 through 7 that corresponds to the location of a second Legend 128KDE card or use the value 15 to indicate that a second card is not available.

210 S3 = 15
Use a value in the range from 0 through 7 that corresponds to the SLOT number that EMULATOR #3 will respond to or use the value 15 to indicate that EMULATOR #3 will not be implemented.

220 D3 = 15
Use the value 1 or 2 that corresponds to the DRIVE number that EMULATOR #3 will respond to or use the value 15 to indicate that EMULATOR #3 will not be implemented.

230 K3 = 15
Use a value in the range from 0 through 7 that corresponds to the location of a third Legend 128KDE card or use the value 15 to indicate that a third card is not available.

280 K4 = 15
Use a value in the range from 1 through 7 that corresponds to the location of a fourth Legend 128KDE card or use the value 15 to indicate that a fourth card is not available.

Disk Emulator (con't.)

SAVE this custom <Turnkey> program onto another diskette for testing-

SAVE TURNKEY <Press RETURN>

The DISK EMULATOR 4.0 file can also be transferred onto another diskette by BLOADing it into memory and then-

BSAVE DISK EMULATOR 4.0, A$2000, L$A00 <Press RETURN>

Listing 1

```
20  REM ************************
22  REM *                      *
24  REM * < TURNKEY EMULATOR > *
26  REM *   MICHAEL MC LAREN   *
28  REM *  (C) COPYRIGHT 1981  *
30  REM
32  REM  LEGEND INDUSTRIES, LTD.
34  REM        P.O. BOX 112
36  REM    PONTIAC, MI.   48056
40  REM *                      *
42  REM ************************
50  HOME : PRINT "WAIT"
55 :
100 RC = 00: REM ROM CARD (0 DEFAULT NO CARD)
110 S1 = 06: REM DEM#1 EMULATES SLOT 6
120 D1 = 02: REM DEM#1 EMULATES DRIVE 2
130 K1 = 00: REM 1ST 128k DE SLOT 0
150 :
160 S2 = 15: REM DEM#2 EMULATES SLOT (15 DEFAULT NO EMULATOR)
170 D2 = 15: REM DEM#2 EMULATES DRIVE (15 DEFAULT NO EMULATOR)
180 K2 = 15: REM 2nd 128KDE SLOT (15 DEFAULT NO CARD)
200 :
210 S3 = 15: REM DEM#3 EMULATES SLOT (15 DEFAULT NO EMULATOR)
220 D3 = 15: REM DEM#3 EMULATES DRIVE (15 DEFAULT NO EMULATOR)
230 K3 = 15: REM DEM#3 128KDE SLOT (15 DEFAULT NO CARD)
250 :
260 S4 = 15: REM DEM#4 EMULATES SLOT (15 DEFAULT NO EMULATOR)
270 D4 = 15: REM DEM#4 EMULATES DRIVE (15 DEFAULT NO EMULATOR)
280 K4 = 15: REM DEM#4 128KDE SLOT (15 DEFAULT NO CARD)
285 :
290  PRINT  CHR$ (4);"BLOAD DISK EMULATOR 4.0"
295 :
300  POKE 9697,RC * 16: REM ROM FIRMWARE SLOT
310  POKE 9704,S1 * 16: REM DEM#1 EMULATED SLOT
320  POKE 9705,D1 * 01: REM DEM#1 EMULATED DRIVE
330  POKE 9712,K1 * 16: REM DEM#1 128KDE SLOT
340  POKE 9713,K1 * 16: REM DEM#1 128KDE SLOT
350 :
360  POKE 9706,S2 * 16: REM DEM#2 EMULATED SLOT
370  POKE 9707,D2 * 01: REM DEM#2 EMULATED DRIVE
380  POKE 9714,K2 * 16: REM DEM#2 128KDE SLOT
390  POKE 9715,K2 * 16: REM DEM#2 128KDE SLOT
400 :
410  POKE 9708,S3 * 16: REM DEM#3 EMULATED SLOT
420  POKE 9709,D3 * 01: REM DEM#3 EMULATED DRIVE
430  POKE 9716,K3 * 16: REM DEM#3 128KDE SLOT
440  POKE 9717,K3 * 16: REM DEM#3 128KDE SLOT
450 :
460  POKE 9710,S4 * 16: REM DEM#4 EMULATED SLOT
470  POKE 9711,D4 * 01: REM DEM#4 EMULATED DRIVE
480  POKE 9718,K4 * 16: REM DEM#4 128KDE SLOT
490  POKE 9719,K4 * 16: REM DEM#4 128KDE SLOT
495 :
500  CALL 8960: REM CALL INSTALL ROUTINE
510 :
520  PRINT "DONE....."
540  PRINT CHR$ (4);"CATALOG,S";S!;",D";D1
```

Disk Emulator (con't.)

Disk Emulator is the perfect solution for those applications that require a lot of timely disk access. The user application program and its associated disk files can be copied into the emulated disk and run from there with a speed improvement of approximately 300% on all of the disk access to the emulated disk.

To gain access to the emulated disk (or disks) simply use the optional slot and drive parameters of the emulated disk in one of the DOS commands. For example, if emulator #1 emulates Slot 5 / Drive 1 then type in the following command.

```
>CATALOG, S5, D1   <Press RETURN>
```

Once the emulated disk has been accessed it remains the default disk until another slot and drive is used in a DOS command.

The emulated disk Directory does not contain any files when you initially install the DISK EMULATOR. You may copy single files from floppy disk onto the emulated disk by using the FID program on the DOS 3.3 System Master disk or you can copy an entire floppy disk onto the emulated disk with the COPY or COPYA programs on the System Master disk.

Disk Emulator provides two new DOS commands for each of the four emulated disks. The <Mount> command copies the floppy diskette in the Slot and Drive that you specify into the emulator that you specify. This command allows the user to mount his application software into the emulator quickly, usually in less than 18 seconds.

Command	Function
.M1,S6,D1	Copy the floppy diskette in Slot 6 / Drive 1 into Emulator #1.
.M2,S6,D2	Copy the floppy diskette in Slot 6 / Drive 2 into Emulator #2.
.M3,S6,D1	Copy the floppy diskette in Slot 6 / Drive 1 into Emulator #3.
.M4,S6,D1	Copy the floppy diskette in Slot 6 / Drive 1 into Emulator #4.

The Mount command is a true DOS command and it can be used in immediate mode by typing it in at the keyboard or defered mode from within a program by using the standard Apple DOS print control-D syntax, example-

```
800 PRINT CHR$(4);".M1 ,S6 ,D1"
```

The Mount command copies only those tracks on the floppy diskette that the emulator can store. If you are emulating a full-disk then you will mount tracks 3 through 34 or tracks 7 through 34 into the emulated disk, depending on whether or not Bank 0 is used to contain the missing language in ROM.

Disk Emulator (con't.)

When you emulate anything less than a full-disk (32 tracks), you must insure that the disk files on the floppy diskette that you mount are all located within the specified range of tracks.

The SPECIAL FORMAT program on your Legend diskette is a utility program which helps you set up partial disks especially for use with the Mount and Update commands. Use of the SPECIAL FORMAT program is described later in this manual.

The <Update> command copies the contents of the emulator that you specify onto floppy disk in the Slot and Drive that you specify. This is useful when any data type files have been modified during the operation of the users application program. The Update command allows the user to record the changes on the emulated disk permanently on a floppy diskette.

Command	Function
.U1,S6,D1	Copy Emulator #1 onto the floppy diskette in Slot 6 / Drive 1.
.U2,S6,D2	Copy Emulator #2 onto the floppy diskette in Slot 6 / Drive 2.
.U3,S6,D1	Copy Emulator #3 onto the floppy diskette in Slot 6 / Drive 1.
.U4,S6,D2	Copy Emulator #4 onto the floppy diskette in Slot 6 / Drive 2.

The Update command is a true DOS command and it can be used in immediate mode by typing it in at the keyboard or defered mode from within a program by using the standard Apple DOS print control-D syntax, example-

```
900 PRINT CHR$(4);".U1 ,S6 ,D1"
```

The Update command copies only those tracks onto the floppy diskette that are contained on the emulated disk. If the emulated disk is a full-disk then tracks 3 through 34 or tracks 7 through 34 will be written onto the floppy diskette, depending on wether or not BASIC is contained in Bank Ø on the 128KDE card.

** CAUTION **

The <Mount> and <Update> commands should be used with extreme care. The user should format several floppy diskettes as 'Partial' diskettes with the SPECIAL FORMAT program when the Disk Emulator is set-up to emulate anything less than a full-disk (tracks 3 through 34). Any files that the user copies onto these diskettes will be placed in the correct area of the diskette.

DISK EMULATOR 4.0

```
0800          101  ************************************************
0800          102  *                                              *
0800          103  * <DISK EMULATOR 4.0>                          *
0800          104  * by Michael D. McLaren (Version 11/02/81)     *
0800          105  *(C) Copyright 1981-Legend Industries, Ltd.    *
0800          106  *          ALL RIGHTS RESERVED                 *
0800          107  *                                              *
0800          108  ************************************************
0800          109  ;
0008          110  PTRL      EPZ $08
003E          111  ADRL      EPZ $3E
0048          112  IOB       EPZ $48
0800          113  ;
BF41          114  DOSON     EQU $BF41
BF65          115  DOSOFF    EQU $BF65
0800          116  ;
9C00          117            ORG $9C00
9C00          118            OBJ $0800
9C00          119  ;
9C00 8CF806   120  EMTEST1   STY $6F8          ;.................
9C03 A206     121            LDX ##$06        ;Index for 4 test
9C05 88       122            DEY              ;Index for IOB #
9C06 B148     123  DEMTEST   LDA (IOB),Y      ;.................
9C08 DDE89C   124            CMP EMSLOT,X     ;Same as EM#X nbr
9C0B D009     125            BNE TESTNXT      ;No,try next emul
9C0D C8       126            INY              ;Index for IOB d#
9C0E B148     127            LDA (IOB),Y      ;.................
9C10 DDE99C   128            CMP EMSLOT+1,X   ;Same as EM#X dr#
9C13 F008     129            BEQ EMULATE      ;Yes,then emulate
9C15 88       130            DEY              ;.................
9C16 CA       131  TESTNXT   DEX              ;Index next EM pa
9C17 CA       132            DEX              ;Tested all 3 par
9C18 10EC     133            BPL DEMTEST      ;No,test next EM.
9C1A 4C09BD   134            JMP $BD09        ;Else,ret to RWTS
9C1D          135  ;
9C1D A00C     136  EMULATE   LDY ##$0C        ;Index for IOB co
9C1F B148     137            LDA (IOB),Y      ;.................
9C21 F06D     138            BEQ EXIT1        ;Yes?,exit emulat
9C23 C904     139            CMP ##$04        ;Is it format com
9C25 F069     140            BEQ EXIT1        ;Yes then exit em
9C27 8DE29C   141            STA IOBCMD       ;Save IOB command
9C2A A004     142            LDY ##$04        ;Index for IOB t#
9C2C B148     143            LDA (IOB),Y      ;.................
9C2E DDF89C   144            CMP TRKLMT,X     ;Is it tr 0,1 or2
9C31 905D     145            BCC EXIT1        ;Yes then exit em
9C33 DDF99C   146            CMP TRKLMT+1,X   ;Beyond last trk?
9C36 B058     147            BCS EXIT1        ;Yes then exit em
9C38 E902     148            SBC ##$02        ;64KC add offset
9C3A 4B       149            PHA              ;Save new trk val
9C3B C8       150            INY              ;Index for IOB se
9C3C B148     151            LDA (IOB),Y      ;.................
9C3E 8DE19C   152            STA SECTOR       ;Save pg.# $00-0F
9C41 68       153            PLA              ;Retreive tr valu
9C42 A8       154            TAY              ;Save new tr valu
```



```
9C43 20B79C   155            JSR SAVLANG      ;Save lan ID byte
9C46 20929C   156            JSR ADRCALC      ;Set up 128KC add
9C49 A008     157            LDY ##$08        ;Index for IOB bu
9C4B B148     158            LDA (IOB),Y      ;Get buffer lo by
9C4D 853E     159            STA ADRL         ;.................
9C4F C8       160            INY              ;Index for buf hi
9C50 B148     161            LDA (IOB),Y      ;.................
9C52 853F     162            STA ADRL+1       ;.................
9C54 A000     163            LDY ##$00        ;Init Y index reg
9C56 8408     164            STY PTRL         ;.................
9C58 ADE29C   165            LDA IOBCMD       ;.................
9C5B 4A       166            LSR              ;Is it RD command
9C5C B013     167            BCS RDSECT       ;YES read sector
9C5E EAEAEA   168            HEX EAEAEA      Future DEM xtnsion
9C61 BD83C0   169            LDA $C083,X      ;Turn on 128KC
9C64 BD83C0   170            LDA $C083,X      ;Write enable car
9C67 B13E     171  WRSECT    LDA (ADRL),Y     ;.................
9C69 9108     172            STA (PTRL),Y     ;.................
9C6B C8       173            INY              ;All 256 bytes mov
9C6C D0F9     174            BNE WRSECT       ;If not,move more
9C6E F00C     175            BEQ EXIT         ;Else do norm ext
9C70 EA       176            HEX EA           ;.................
9C71 BD83C0   177  RDSECT    LDA $C083,X      ;Select 128KC crd
9C74 B108     178  RDSECT1   LDA (PTRL),Y     ;.................
9C76 913E     179            STA (ADRL),Y     ;.................
9C78 C8       180            INY              ;All 256 bytes mo
9C79 D0F9     181            BNE RDSECT1      ;If not,move more
9C7B EA       182            HEX EA           ;.................
9C7C          183  ;
9C7C BD82C0   184  EXIT      LDA $C082,X      ;Turn off 128KC
9C7F 98       185            TYA              ;Load A w zero va
9C80 9D84C0   186            STA $C084,X      ;Select bank# 0
9C83 A00E     187            LDY ##$0E        ;Index for IOB vl
9C85 ADE39C   188            LDA VOLUME       ;Set vol nbr 254
9C88 9148     189            STA (IOB),Y      ;.................
9C8A ADE09C   190            LDA LANGID       ;Get lang ID byte
9C8D 20B2A5   191            JSR $A5B2        ;DOS ROM switch r
9C90 18       192  EXIT1     CLC              ;Indicates no err
9C91 60       193            RTS              ;Return to call r
9C92          194  ;
9C92          195  ;ADRCALC- Y register - Track value %000XXXXX
9C92          196  ;           Indicates one of 32 (4K) tracks
9C92          197  ;         X register - 128KC index %000000XX
9C92          198  ;           Indicates one of three EMULATORS
9C92          199  ;         SECTOR     - Sector nbr. %0000XXXX
9C92          200  ;           Indicates page number $00 to $0F
9C92          201  ;
9C92          202  ; ADRCALC subroutine returns with hi-byte
9C92          203  ; of 128Kc add. in PTRL+1 ptr.Register
9C92          204  ;X contains index for correct 128KC slot.
9C92          205  ;
9C92 98       206  ADRALC    TYA              ;Retrieve track v
9C93 2910     207            AND #%    ;1st or 2nd card?
9C95 F001     208            BEQ CONT1        ;Branch if card#1
9C97 E8       209            INX              ;Index for card#2
```

```
9C98 BDF09C   210 CONT1     LDA NDXTBL,X      ;Get 128KC slot
9C9B AA       211           TAX               ;Put in X index r
9C9C 98       212           TYA               ;Retrieve trk val
9C9D 291C     213           AND #%            ;Mask for bank no
9C9F 4A       214           LSR               ;.................
9CA0 4A       215           LSR               ;.................
9CA1 9D84C0   216           STA $C084,X       ;Set correct bank
9CA4 98       217           TYA               ;Retrieve trk val
9CA5 2903     218           AND #%            ;Mask for 4K bnks
9CA7 A8       219           TAY               ;Index for 128KC
9CA8 D004     220           BNE CONT2         ;Branch if not 1
9CAA 8A       221           TXA               ;.................
9CAB 6908     222           ADC #$08          ;1st 4K sub-bank
9CAD AA       223           TAX               ;.................
9CAE B9E49C   224  CONT2     LDA ADRTBL,Y      ;Get 128Kc add hi
9CB1 6DE19C   225           ADC SECTOR        ;Add sector 00-0F
9CB4 8509     226           STA PTRL+1        ;Save pointer hi-b
9CB6 60       227           RTS               ;Return to call r
9CB7          228 ;
9CB7 AD00E0   229  SAVLANG   LDA $E000        ;.................
9CBA 8DE09C   230           STA LANGID        ;.................
9CBD AD82C0   231           LDA $C082         ;Wr.Prot.slot #0.
9CC0 AD81C0   232           LDA $C081         ;Turn off slot #0
9CC3 60       233           RTS               ;Return to call r
9CC4          234 ;
9CC4 B13E     235  WRDEM     LDA (ADRL),Y      ;.................
9CC6 9900BD   236           STA $BD00,Y       ;.................
9CC9 C8       237           INY               ;.................
9CCA D0F8     238           BNE WRDEM         ;.................
9CCC 4C65BF   239           JMP DOSOFF        ;.................
9CCF 2065BF   240  RDDEM     JSR DOSOFF        ;.................
9CD2 BD83C0   241           LDA $C083,X       ;Select 128KCard
9CD5 B108     242  RDDEM1    LDA (PTRL),Y      ;.................
9CD7 9900BD   243           STA $BD00,Y       ;.................
9CDA C8       244           INY               ;.................
9CDB D0F8     245           BNE RDDEM1        ;.................
9CDD 4C41BF   246           JMP DOSON         ;.................
9CE0          247 ;
9CE0 00       248  LANGID    HEX 00            ;.................
9CE1 00       249  SECTOR    HEX 00            ;.................
9CE2 00       250  IOBCMD    HEX 00            ;.................
9CE3 FE       251  VOLUME    HEX FE            ;Vol.# found 254.
9CE4          252 ;
9CE4 D0       253  ADRTBL    HEX D0           ;1st 4K sub-bank
9CE5 D0       254            HEX D0           ;2nd 4K sub-bank
9CE6 E0       255            HEX E0           ;2nd 4K sub-bank
9CE7 F0       256            HEX F0           ;2nd 4K sub-bank
9CE8          257 ;
9CE8          258 ;Hardware config.block.
9CE8          259 ;
9CE8 5001     260  EMSLOT    HEX 5001        ;EM#1 slot/dr.#
9CEA FFFF     261            HEX FFFF        ;EM#2 slot/dr.#
9CEC FFFF     262            HEX FFFF        ;EM#3 slot/dr.#
9CEE FFFF     263            HEX FFFF        ;EM#4 slot/dr.#
9CF0 2040     264  NDXTBL    HEX 2040        ;EM#1 1st/2nd i
9CF2 FFFF     265            HEX FFFF        ;EM#2 1st/2nd i
```



```
9CF4 FFFF     266            HEX FFFF        ;EM#3 1st/2nd
9CF6 FFFF     267            HEX FFFF        ;EM#4 1st/2nd
9CF8 0323     268  TRKLMT    HEX 0323        ;EM#1 beg/end
9CFA 0323     269            HEX 0323        ;EM#2 beg/end
9CFC 0323     270            HEX 0323        ;EM#3 beg/end
9CFE 0323     271            HEX 0323        ;EM#4 beg/end
              272            END

**** END OF ASSEMBLY
```

SPECIAL FORMAT program

The SPECIAL FORMAT program allows the user to create special 'partial' diskettes which may be used for the storage of disk files. The SPECIAL FORMAT program will initialize a blank diskette and then mark the Volume Table Of Contents (VTOC) in the Directory of the diskette to show that only the tracks in the specified range are available for the storage of disk files.

Once the floppy diskette has been formatted and marked as a 'partial disk' the user may copy individual files onto it with the Apple FID program on the System Master disk. All files will be written onto the disk in the specified range of tracks. This insures that the <Mount> and <Update> commands will move a valid range of tracks to and from the emulated disk.

To use the SPECIAL FORMAT program simply insert the Legend disk into the drive and type-

RUN SPECIAL FORMAT (press RETURN)

The program clears the screen and then prompts the user with three important set-up questions.

GREETING PROGRAM -

Type in the name of the greeting program you intend to use on the new diskette. The SPECIAL FORMAT program does not place the greeting program on the new disk, it simply places the greeting program name into the proper area of DOS on the new disk.

BEGINNING TRACK (3 OR 7) -

If the 128KDE card is installed in slot Ø and no Integer or Applesoft FIRMWARE card is installed in the computer, then Disk Emulator will retain the first 16K Bank of RAM on the card for the BASIC language missing in ROM on the Apple mother board. This prevents the use of the first four tracks on the emulated disk, tracks 3 through 6, so type in the value '7' in responce to this question.

If the 128KDE card is installed in any slot other than slot Ø OR if an Integer or Applesoft FIRMWARE card is installed, then type in the value '3' in responce to this question.

ENDING TRACK (18 OR 34) -

If a 64KC card is installed in the system and you wish to format a "half-disk" using this card, type in the value '18'. If a 128KDE card is installed in the system and you wish to emulate a "full-disk", type in the value '34'.

INSERT BLANK DISK AND PRESS <RETURN>

Insert a blank disk into the drive and press <RETURN> or any other key. When the disk has been initialized, the user can place the greeting program and other application programs onto it.

Listing 2

```
20  REM ************************
22  REM *                      *
24  REM * < FLOPPY FORMATTER > *
26  REM *   MICHAEL MC LAREN   *
28  REM *  (C) COPYRIGHT 1981  *
30  REM
32  REM  LEGEND INDUSTRIES, LTD.
34  REM        P.O. BOX 112
36  REM    PONTIAC, MI.   48056
40  REM *                      *
42  REM ************************
44 :
50  HOME : PRINT "WAIT"
55 :
100  FOR I = 1 TO 40: PRINT "=";: NEXT
110  PRINT "       SPECIAL DISK FORMAT PROGRAM"
120  PRINT " (C) COPYRIGHT 1981, LEGEND INDUSTRIES"
130  FOR I = 1 TO 40: PRINT "=";: NEXT : PRINT
140 :
150  VTAB 7: INPUT "GREETING PROGRAM - ";GN$
170  IF GN$ = "" THEN GN$ = "HELLO"
180 :
200  VTAB 9: INPUT "BEGINNING TRACK (3 OR 7) - ";BT$
210 BT = VAL (BT$)
220 IF BT < 3 OR BT > 11 THEN PRINT "": GOTO 200
230 :
250  VTAB 11: INPUT "ENDING TRACK (18 OR 34) - ";ET$
260 ET = VAL (ET$)
270  IF ET < > 18 AND ET < > 34 THEN PRINT "": GOTO 250
280 :
300  VTAB 14: PRINT "INSERT BLANK DISK AND PRESS <RETURN> ";
310  GET A$: PRINT
320 :
350  POKE  - 20813,BT * 4: REM BEGIN TRACK
360  POKE  - 20811,(ET + 11) * 4: REM END TRACK
370 :
380  PRINT  CHR$ (4);"INIT";GN$
390  PRINT  CHR$ (4);"DELETE";GN$
400 :
410  POKE  - 20813,12: POKE  - 20811,140
```


Firmware Selector

The Firmware Selector utility program was written especially for Apple II users that own either an Integer or Applesoft FIRMWARE card as well as the Legend 64KC or 128KDE RAM card. At last, the power and flexibility of a RAM card in slot 0 for Pascal, Fortran, etc. and the convenience of a FIRMWARE card in some other slot for Integer BASIC or Applesoft BASIC.

The Firmware Selector program may be used to modify the DOS on your DOS 3.3 diskettes to recognize a FIRMWARE card installed in any slot inside the Apple. A DOS 3.3 diskette that has been modified using the Firmware Selector program will, when booted, recognize and control the selection and deselection of the FIRMWARE card installed in the slot that you specified.

The Firmware Selector program is easy to use. Simply RUN FIRMWARE SELECTOR on the Legend diskette. The program will prompt you to insert a standard DOS 3.3 diskette into the drive and then press <RETURN>. Be sure to remove the write protect sticker on the DOS 3.3 diskette before inserting it into the drive. The program will read this diskette and display the current slot number that the DOS on this diskette will use when referencing a FIRMWARE card (normally slot 0) and prompt you for a new slot number. Type in a slot number in the range from 0 through 7 that corresponds to the new location of the FIRMWARE card in your system (slot 4 is a good choice). The program then modifies two bytes in the ROM switching routine (ROMSW) within DOS on this disk.

That's all there is to it. The modified diskette, when booted, will automatically recognize your FIRMWARE card installed in slot 4 or any other slot that you selected and your Pascal disks, when booted, will work fine with the Legend RAM card installed in slot 0.

```
20  REM ************************
22  REM *                      *
24  REM * FIRMWARE SELECT PROG *
26  REM * (MICHAEL D. MCLAREN) *
30  REM *                      *
32  REM *  (C) COPYRIGHT 1981  *
34  REM * ALL RIGHTS RESERVED! *
38  REM
40  REM  LEGEND INDUSTRIES LTD.
42  REM        P.O.BOX 112
44  REM     PONTIAC, MI. 48056
46  REM
48  REM *                      *
50  REM ************************
55 :
100  TEXT : HOME
110  FOR I = 1 TO 40: PRINT "=";: NEXT
120  PRINT "      LEGEND FIRMWARE SELECT PROGRAM"
130  PRINT "(C)OPYRIGHT 1981 LEGEND INDUSTRIES LTD."
140  FOR I = 1 TO 40: PRINT "=";: NEXT : PRINT
150 :
160  GOSUB 460
170 :
180  PRINT "  THIS PROGRAM WILL MODIFY THE LANGUAGE ";
190  PRINT "SWITCHING ROUTINE (ROMSW) WITHIN DOS ON ";
200  PRINT "YOUR DOS 3.3 DISKETTES TO AUTOMATICALLY ";
210  PRINT "RECOGNIZE AN INTEGER OR APPLESOFT FIRM- ";
220  PRINT "WARE CARD INSTALLED IN SLOT 0 THRU 7."
230  PRINT : PRINT
240 :
250  PRINT " INSERT TARGET DISK AND PRESS <RETURN>";: GET A$: PRINT
260  CALL 769: REM READ TRK.1 SCT.4
270  POKE 34,4: HOME : POKE 34,0: PRINT
280 :
290  PRINT " THE DOS ON THIS DISKETTE, WHEN BOOTED, ";
300  PRINT " WILL RECOGNIZE AN INTEGER OR APPLESOFT ";
310  PRINT " FIRMWARE CARD INSTALLED IN SLOT NBR. ";
320  INVERSE : PRINT ( PEEK (6328) - 128) / 16: NORMAL
330 :
340  VTAB 10: PRINT " NEW FIRMWARE CARD SLOT NUMBER (0-7)? ";: GET SL
350 SL =  VAL (SL$): IF SL < 0 OR SL > 7 THEN 340
360  INVERSE : PRINT SL$: NORMAL
370 :
380  POKE 6328,128 + SL * 16: POKE 6336,129 + SL * 16
390  CALL 773: REM WRITE TRK.1 SCT.4
400 :
410  PRINT : PRINT " UPDATE ANOTHER DISKETTE ?";: GET A$
430  PRINT : IF A$ = "Y" THEN 100
440  END
450 :
460  RESTORE : FOR I = 768 TO 831: READ D: POKE I,D: NEXT : RETURN
490 :
500  DATA 000, 169, 001, 208, 002, 169, 002, 072
510  DATA 032, 227, 003, 132, 072, 133, 073, 160
520  DATA 003, 169, 000, 141, 000, 003, 145, 072
530  DATA 200, 169, 001, 145, 072, 200, 169, 004
540  DATA 145, 072, 160, 008, 169, 000, 145, 072
550  DATA 200, 169, 024, 145, 072, 160, 012, 104
560  DATA 145, 072, 032, 227, 003, 032, 217, 003
570  DATA 144, 005, 169, 255, 141, 000, 003, 096
580 :
```

Legend Industries Ltd.